Modeling the Emergence of Phonology with Multi-Agent Reinforcement Learning

Rujul Gandhi
rujul@mit.edu

Abstract

Remarkable cross-linguistic similarities in how natural language phonologies have evolved presents an interesting modeling problem - can this be replicated with just a computational simulation? Past models and simulations of this evolution process have focused on agents agreeing upon a phonology, but the pressure of needing to communicate efficiently about the world does not factor into these models. This project presents a communication-based approach to the evolution of phonological patterns by proposing experiments in an emergent communication setting. We develop a multi-agent reinforcement learning architecture for auto-encoding simple inputs and transmitting communication as sequences of phonemes, which provides an environment and base neural network model to simulate two key phenomena - the selection of phonemes, and the emergence of sequence constraints - as a result of parameters such as noise, perceptual distance, or articulatory effort.

1 Introduction

At the surface level, human communication is a continuous stream of sounds. As natural languages begin to map sequences of sounds, or 'words', to concepts, patterns emerge within the sounds and their sequences. These patterns, which govern what is considered a permissible 'word' in the language vocabulary, form the phonology of the language.Although languages differ in the specific rules or constraints that form their phonology, there are some remarkably widespread patterns and phenomena.

Two key things form a 'phonology' for a natural language. It has a 'fixed inventory of distinctive segments' and there must exist any 'sequential constraint(s) on segments' (Hyman, 2008).

The first category concerns the sounds which become contrastive segments of a particular language. The best example of this is the statistical survey of world phoneme inventories in the UCLA Phonological Segment Inventory Database (UPSID). This survey compiles contrastive segment inventories for 451 natural languages. Of 921 attested speech sounds in this dataset (De Boer, 2000), the average number of sounds in a phoneme inventory is between 20-37 (Lass, 1986). The selection of which sounds are contrastive is not at random, with (Lass, 1986) finding that just within consonants, over 90% of 317 languages surveyed included simple nasal and stop sounds (/m/, /t/, /n/, /k/) with that percentage decaying down to just about 20% of languages including the 25^{th} most common consonant sound /v/. Similar studies have been done for vowels (Liljencrants and Lindblom, 1972).

The second category of constraints in a phonology is sequential ones. For instance, languages are less likely to contain syllables with a coda than simple V or CV syllables (De Boer, 2000). Another example of a widespread pattern is languages following the Sonority Sequencing Principle in syllabification - where consonants are arranged in an increasing or decreasing order of sonority, around the central sonority peak provided by the vowel. As an example, consonant cluster preferences at the start of a syllable would be [pl] >[lp] whereas at the end of a syllable it would be [lp] >[pl], owing to /l/ being more sonorant than /p/ (De Boer, 2000).

A clear experimental question arising from these descriptive studies is whether it is possible to simulate the evolution of a natural-seeming phonology, through identifying and modeling the factors that constrain it.

This project is intended to be a preliminary inquiry into how the patterns formed by neural net-

work agents optimizing for efficient communication compare to attested patterns in the natural languages of today.

2 Background and Related Work

2.1 Modeling Phonology

In explaining phonology from a phonetic perspective, a persistent idea is that there are opposing weights of wanting to place your phonemes far apart in acoustic, or perceptual space, and minimize the articulatory effort of switching between phonemes while doing so. This potential tradeoff leads to an interesting modeling problem, similar in some ways to studies modeling an information bottleneck in developing semantic spaces (Zaslavsky et al., 2018).

Modeling patterns of phonology through trying to model this balance is a well-studied problem. Some early papers passed single vowels between agents, where the goal of the agents was to imitate each other as well as keep an evolutionary advantage (Glotin 1995, as cited by (De Boer, 2000)). (De Boer, 2000) expands significantly upon these initial efforts, equipping a group of agents with models of human perception and articulation for vowels. De Boer's experiments find natural-language-like patterns arising in the phonology evolved collectively by these agents.

De Boer's methodology is meticulous in that it uses well-accepted models of perception and articulation. Interestingly, though, the agents in De Boer's simulation operate with the main goal of imitating each other as well as possible. Intuitively, as language evolves, we expect a balance between wanting to converge to a set of representations (i.e. imitate each other) and wanting to communicate about the world. Importantly, the modeling simulations of phonology thus far do not explicitly tie model utterances to a task or an external world. It is likely that in the case of human language, it is influenced by the reasons for which we communicate - for instance, collaborating on tasks, or communicating ideas. For this reason, studies of agents communicating about events or environments external to themselves is of interest in modeling natural language.

2.2 Emergent Communication

Emergent communication in multi-agent reinforcement learning (MARL) settings has recently been studied for giving rise to apparent natural-language-like patterns, as observed in the context of referential games (Lazaridou et al., 2016). In Emergent Communication experiments, neural network agents may communicate with each other to achieve interrelated goals. The goals may be cooperative or competitive, but since both agents' rewards are tied to the goal and their abilities are often different (for instance, one agent may have the ability to observe the environment while only the other agent has the ability to actually take actions in the environment), they are induced to converge upon a meaningful communication system. Outside of the application of EC to robotics and computer science, it is interesting to linguists for two reasons.

First, there is the question of whether applying natural-language-processing techniques or implementing factors from human language production and perception can make this communication between agents more efficient, quick to learn, or robust. A couple of recent effort in this direction have been introducing multidimensional semantic spaces to discrete emergent communication (Tucker et al., 2021) or constraining the communication through making the communication channel noisy, as it often is in the real world leading to inference by rational speaker and listeners. The second linguistic contribution of EC is that probing the nature of the communication between neural network agents to compare it to natural-language often shows the differences between the two and the ways to induce EC to be more natural-language-like (Kottur et al., 2017). Being able to run a semi-controlled experiment like this could be an interesting modeling tool in addition to current theoretical frameworks and rule-based models of linguistics.

This framework is interesting for studying the evolution of phonology since the agents have the incentive to converge on an efficient communication pattern. If the task is simply the reconstruction of an input (i.e. one agent sees the input and emits communication, the other agent decodes from the communication), the agents are incentivized to converge on a vocabulary. Since the phonology is best studied at the level of creating words, the contents of such a vocabulary are ideal to model emerging phonology.

3 Model Architecture

The model architecture essential includes two collaborating agents, coded as part of a single network, operating within an environment that selects inputs and determines rewards based on the model outputs.

3.1 Environment and Task

We define a simple 'environment' for the agents where the goal is to essentially auto-encode one-dimensional vectors. The environment is initialized with a concept pool C, containing a set number of 'concepts', where each concept is a distinct 1D vector of length `input_dim` that consists of 1s and 0s in some order.

The agents are a `speaker` and `listener`. In each episode, a single input concept c_i is drawn, which is visible to the `speaker` agent. The speaker is able to send communication of a fixed length to the `listener`, using which the `listener` guesses a concept $\hat{c}_i$. Both agents get a positive reward if $\hat{c}_i = c_i$ and a reward of 0 if $\hat{c}_i \neq c_i$. The agents are therefore functioning as the encoder and decoder of a modified auto-encoder, where the latent representation is the communication that gets passed from `speaker` to `listener`.

The choice of an auto-encoder environment over a reference game environment is to avoid confounding factors like the emergence of compositionality or communication about features of the input. In a reference game, the `listener` typically has to discriminate between target and distractor concepts from different 'categories.' This can lead to an interesting semantic structure, but it is not the focus of this experiment. In human language, breaking down meaning to the level of individual phonemes is rare if even possible, so we are intentionally avoiding it here. The auto-encoder task incentivizes the models to develop a vocabulary that assigns distinct, potentially unrelated communication sequences to each concept in C.

3.2 Models

The model architecture is adapted from the reference implementation for `ic3net`(Singh et al., 2018).[1], modified for phoneme-sequence communication. The network itself is composed of two

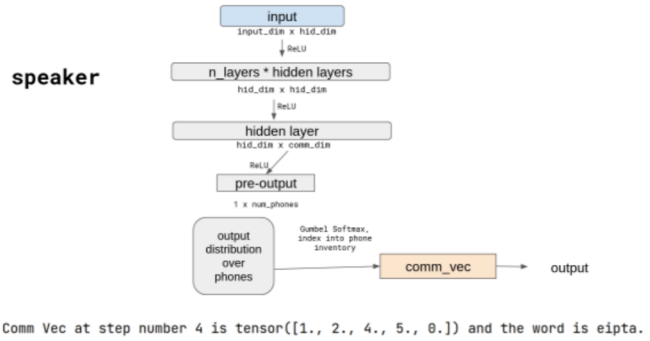

Figure 1: Speaker architecture from an input c_i of size `input_dim|` to output of size `comm_dim|`, which represents the output word as a sequence of indices into the phone inventory.

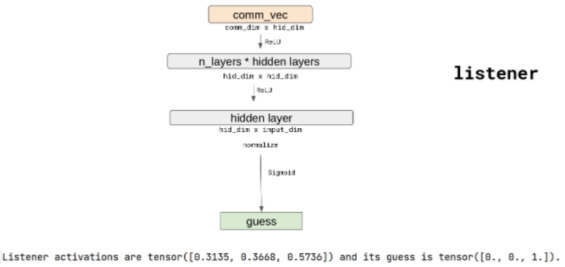

Figure 2: Listener architecture from an input communication vector of size `comm_dim|` to output of size `input_dim|`, which represents the guess $\hat{c}_i$.

smaller Multi-Layer Perceptron (MLP) models, the `speaker` and the `listener`. Each consists of a linear layer that transforms the input into a latent representation, two hidden linear layers with ReLU activation, and an output layer. For the speaker, the input is the input concept vector c_i chosen by the environment, and the output at every time-step is a distribution over the potential phones in the phone inventory.

The output of the `speaker` is converted into a single chosen phone by applying the Gumbel Softmax function to it. This results in a one-hot vector that has a value of 1 at the index of the chosen phone, and can be used to select the correct phone embedding from the inventory.

One version of the model implemented communication through a noisy channel. Gaussian noise is added to the embedding for the phone selected by

[1] `https://doi.org/10.48550/arxiv.1812.09755` Code available under the MIT License.

`speaker`. The result is then compared to all the phone embeddings to find the closest one in terms of minimum squared distance. This closest phone is added to the final communication passed to the listener. Noise can be turned off or varied.

The listener, similarly to the speaker, has multiple linear layers that transform the current state of the communication vector into a latent representation. Based on this representation, the speaker outputs a vector of length input_dim + 1, where the first index of this output is a decision about whether or not the listener wants to guess the word in that time step, and the remainder of the vector is the actual guess. The listener's output layer goes through a Sigmoid activation function, thus effectively we are building a layer that classifies each of these input_dim + 1 positions into a probability between 0 and 1. When calculating the reward, the output of the model is rounded to give a vector of 0s and 1s.

3.3 Reward Function

In addition to the reward function that assigns a positive reward for correct guess and zero for incorrect guess, a few different reward functions were tried. One experiment built up the word step by step, so that the speaker emitted one phone per time step until the full word was formed. The listener could decide at each time step whether to guess or not, except the last time-step which forced a guess. Both agents would receive a large positive reward if a correct guess was made, zero or a negative reward for a wrong guess (there was slightly better performance with zero reward for incorrect guesses), and a small positive reward if the listener took no action at one of the intermediate time-steps.

The motivation for this sort of reward function is human behavioural studies showing people regularly inferring words before hearing them completely. It seems that in natural language communication, as we hear each sound, we are narrowing down the state space until we have a confident guess of the word that the speaker intended. Neural network agents doing the same as they converge to a vocabulary would be an ideal result.

3.4 Representing Phones as Vectors

One of the key challenges of this project is the representation of phones in a way that indicates their acoustic and articulatory characteristics. Most theoretical models of phonology treat phonemes as collections of categorical features. Although this representation lends itself well to a vector format, an ideal representation encodes phones in perceptual (or acoustic) and articulatory space in a more general way than categorical features.

	Sonority	Tube Length	Tongue Height	Delayed Release
a	2	1	0	-1
e	2	-1	1	-1
i	2	0	2	-1
w	2	3	2	-1
p	-2	0	0	0.015
t	-2	1	0	0.025
c	-2	2	0	0.030
k	-2	3	0	0.035
?	-2	4	0	0.060
m	0	0	0	-1
s	-1	1	1	-1
l	1	1.3	0	-1
r	1	1.3	1	-1

Table 1: Phone Representations with human-interpretable placeholders and four articulatory-acoustic features.

Eventually, each phone was encoded as a one-dimensional array of key articulatory features. This representation was chosen after reviewing literature in acoustics, as well as through studying spectrograms of the particular sounds of interest.

In a future experiment, it may be interesting to incorporate a model of human perception such as the Maeda model for vowel simulation (Maeda, 1982) to convert these articulatory features directly into acoustic or perceptual ones.

The phone representations are listed in 1. It is important to note that the characters representing phones are simply representations of a set of articulatory features. There is a correlation, but for instance, an experimental result that shows a phone representing a vowel being often selected after one of the phones representing a stop consonant is not necessarily an indication that a particular sequence like 'pa' or 'ti' is favored, but rather that a heavily sonorant, voiced phone is favored after an obstruent with certain features.

4 Discussion

4.1 Success

This project builds the basic architecture required to simulate phonology through emergent communication. In particular, the model encoder successfully outputs a selection of phonemes, either all at once or over consecutive time-steps, and the model decoder converts this into a guess about the contents of the input. The environment correctly assigns a reward based on the decoding. Example interactions are produced in figures 1 and 2.

The code developed in this project includes additional arguments that control sections of the architecture, relevant in later proposed experiments. Additionally, parts of the code can be updated to study their effect on the system.

- `full_words` : This controls whether the speaker must communicate in full words, or outputs a single phoneme at every time step until the full word is created. Consequently, the listener either must wait for the whole word before guessing the input, or can choose whether or not to guess at every time step depending on its confidence level.

- `transition_loss` : This is a loss penalizing articulatory constraints. Specifically, we propose using MSE over the phone embeddings since sounds that are articulated far apart have values far from each other, and also correspond to higher effort in order to articulate them one after the other. Used together with noise, the `transition_loss` can be altered to study the tradeoff between perceptual distance and ease of articulation.

4.2 Shortcomings and Modifications

The key shortcoming is that the current version of the model fails to converge to a shared vocabulary between the agents. We probe a few potential reasons, make modifications, and propose next steps.

One possible reason for the model failing to converge is some non-differentiable transformation in the process of getting from input to reward. This was probed by simplifying the model, removing transition loss entirely as well as removing noise. Gumbel softmax is used in the final activation step of the speaker model since introducing an argmax to retrieve the phone index brought in a non-differentiable function.

The failure is likely not due to the reward function, since multiple different reward functions were tried. The model was trained with a variety of hyperparameters (num_epochs between 10 to 50, epoch_size between 20 to 100, batch_size 1).

4.3 Future Directions

Two key experiments were proposed with this architecture. If the model is able to converge on a vocabulary, the human-interpretable 'word' outputs that it returns can be used in these experiments.

1. The contrastive segment inventory. Currently, the distribution over phones that are used in speaker output communications is a uniform distribution, as shown in 3. This would be expected to change as noise is added and increased. This proposed experiment consists of increasing the noise being applied to the phone embeddings and analyzing resulting phone distributions in the output. The hypothesized result is that a couple of vowels and stop consonants would be the most likely to be represented, with the individual vowels and stop consonants being far from each other perceptually.

2. Sequential constraints. Since transition loss is not applied currently, there is no constraint on how phones are put together. Upon adding it, the hypothesized result is that phones closer together in location and manner of articulation will be frequently emitted together. In this experiment, constraining the number of vowels in the base phone inventory may replicate or probe the effects of a sonority sequencing principle.

Acknowledgements

Thanks to Mycal Tucker for discussions about the concept and guidance on the experimental architecture, and to Jacob Andreas for feedback on the project idea.

References

Bart De Boer. 2000. Self-organization in vowel systems. *Journal of Phonetics*, 27:1–25.

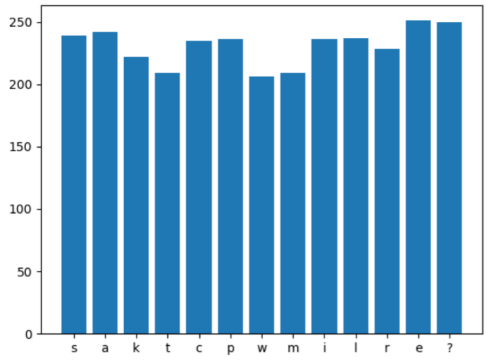

Figure 3: Distribution over phones used in length-3 words, with no noising or transition loss.

Larry Hyman. 2008. Universals in phonology. *Linguistic Review - LINGUIST REV*, 25:83–137.

Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. 2017. Natural language does not emerge 'naturally' in multi-agent dialog.

Roger Lass. 1986. Ian maddieson, patterns of sounds. (cambridge studies in speech science and communication.) cambridge: Cambridge university press, 1984. pp. ix 422. *Journal of Linguistics*, 22(1):200–204.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2016. Multi-agent cooperation and the emergence of (natural) language.

Johan Liljencrants and Björn Lindblom. 1972. Numerical simulation of vowel quality systems: The role of perceptual contrast. *Language*, 48(4):839–862.

Shinji Maeda. 1982. A digital simulation method of the vocal-tract system. *Speech Communication*, 1(3):199–229.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. 2018. Learning when to communicate at scale in multiagent cooperative and competitive tasks.

Mycal Tucker, Huao Li, Siddharth Agrawal, Dana Hughes, Katia P. Sycara, Michael Lewis, and Julie Shah. 2021. Emergent discrete communication in semantic spaces. *CoRR*, abs/2108.01828.

Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. 2018. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942.